

# Introduction to Context Sensitive Scanning with X-KEYSCORE Fingerprints

  
May 2010

Derived From: NSA/CSSM 1-52  
Dated: 20070108  
Declassify On: 20340701

# Opening Question:

How do you find your target's activity in DNI traffic?

# Opening Question:

What if you don't know your targets E-mail address? Or you're trying to find new ones they may be using?

What if the traffic you're interested in doesn't even contain an E-mail address?

What do you do then?



# Opening Question:

You may try to look for keywords or patterns to help find your target.

But how do we scan for keywords in the large volumes of data we see in DNI collection?  
Won't we get too many false hits?

# Context Sensitive Scanning

Context sensitive scanning gives analysts a powerful way to surgically target the traffic you're interested in, by only applying the keywords in the manner in which the analyst intended them to be applied

# Context Sensitive Scanning

- For example, think about these scenarios:
  - “I want to look for documents from Iran that mention a banned item”
  - “I want to look for people doing web searches on Jihad from Kabul”
  - “I want to look for people using Mojahedeen Secrets encryption from an iPhone”
  - “I want to look for documents containing this regular expression”
  - “I want to look for E-mails that mention words from various categories of interest to CP”
- How would you go about targeting those in passive DNI?



# XKS Fingerprints can help!

- Fingerprints are an extremely flexible way to target DNI traffic without the foreknowledge of a strong selector
- They take advantage of X-KEYSCORE's context sensitive scanning engine that has over 70 unique contexts that can be targeted.
- An XKS Fingerprint is simply a meta-data tag that gets applied to a session when a certain criteria is met
- Think of fingerprints as analyst-defined “attributes” of a session

# “There’s an App for that!”

- There are currently almost 10,000 AppIDs and Fingerprints in X-KEYSCORE – the full list is available from the NSA XKS Home Page
- Odds are there may already be a fingerprint for the traffic you’re interested in.
- If not you can easily create your own!



# For example

- I'm an analyst in CT – I want to find anytime Mojahadeen Secrets 2 is seen in DNI Traffic.
- I'm an analyst in CP – I want to find E-mails or Documents relating to the Iranian Nuclear Procurement network
- I'm an analyst in NDIST/NTOC – I want to find traffic from a known botnet

# Use Fingerprints!

**Field Builder**

**AppID (+Fingerprints)**

topic/wmd/iran/iris|

topic/wmd/iran/iris/edi1/chat\_body

topic/wmd/iran/iris/edi1/document\_body

topic/wmd/iran/iris/edi1/email\_body

topic/wmd/iran/iris/edi1/filename

topic/wmd/iran/iris/edi1/url\_path

topic/wmd/iran/iris/edi2

topic/wmd/iran/iris/edi3

**Field Builder**

**AppID (+Fingerprints)**

mojahe

encryption/mojaheden2

encryption/mojaheden2/encodedheader

encryption/mojaheden2/hidden

encryption/mojaheden2/hidden2

encryption/mojaheden2/hidden44

encryption/mojaheden2/secure\_file\_encoded

encryption/mojaheden2/securefile

**Field Builder**

**AppID (+Fingerprints)**

botnet/black|

botnet/blackenergybot/command/die

botnet/blackenergybot/command/flood

botnet/blackenergybot/command/icmp

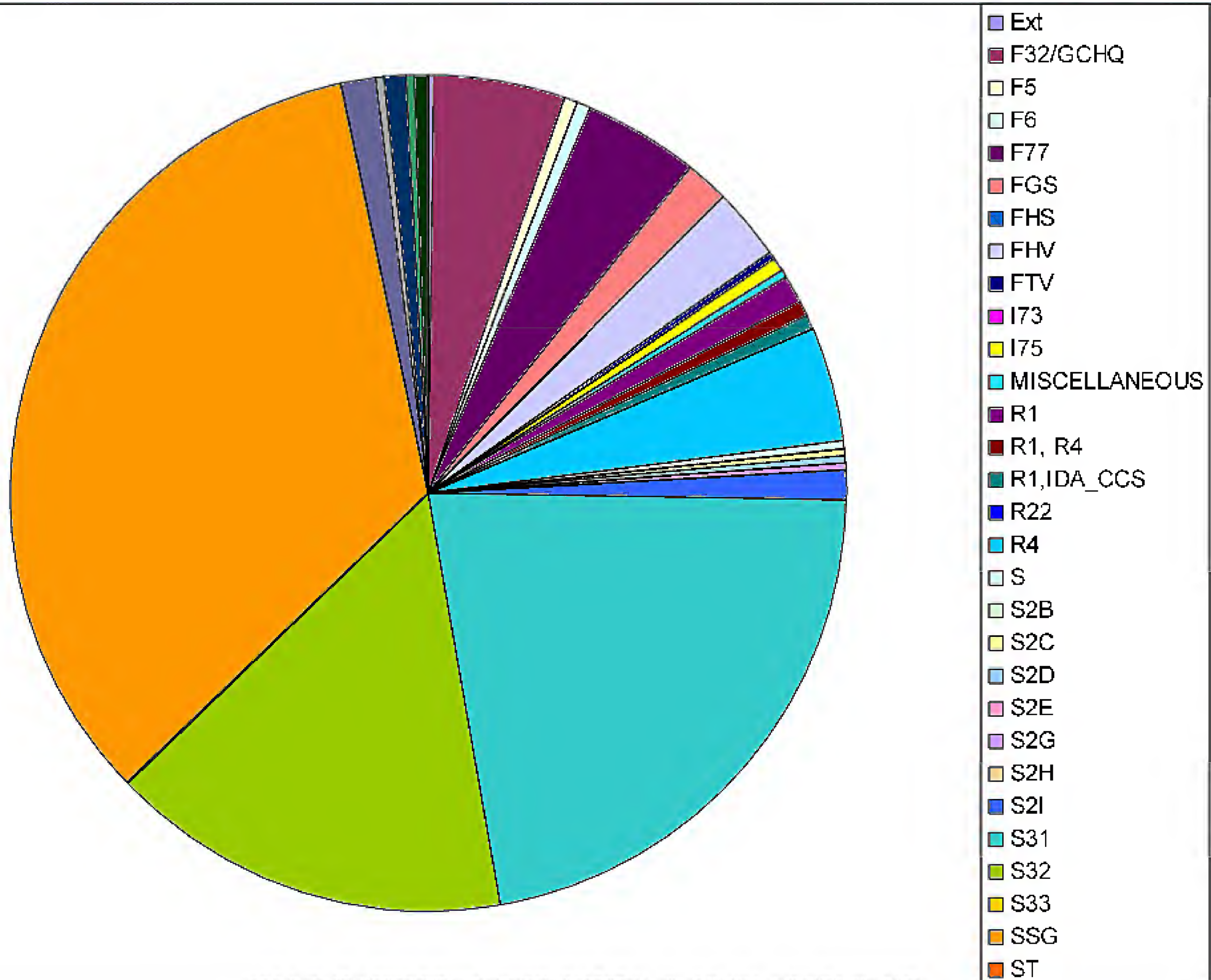
botnet/blackenergybot/command/stop

botnet/blackenergybot/command/syn

botnet/blackenergybot/command/wait



# Who is writing fingerprints?





# Getting Started

- What are the basics of XKS Fingerprints?
- Simple XKS fingerprints are keyword or regular expression based signatures that are evaluated across the data collected and processed by X-KEYSCORE

- |  |  |   |
|--|--|---|
| <p>رسالة خاصة: تفعل ..</p> <p>PM 08:04 - 2008-08-02</p> <p>تاريخ التسجيل: 11 Mar 2007<br/>العضوية: 1,199</p> <p>تفعل ..</p> <p>افراضي تفعل ..</p> <p>### Begin ASRAR El Mojahedeen v2.0 Encrypted Message ###</p> <p>r/RgTzT/ATRhn2E1Zjg1OWQyNWRjMmE2ZTdlNzZmZDhIODUxZWZhMDQ1Mj YwMjViZGU0 ZGYwMjdkMmJmNTA4ZDY2Yjk0MGU2NGNiYjg5MzNjZTc5MThjY2 Y1ZmY6MTgzZDIkYjhjMTE xOGYzYjc1ZDdlMDAxNTQzZmVlNDVlY2YyMGJjYjU2ODkyYjdmY JFjYjAzMWM5ZDQ2OWFIMzg 4NThhM2I1Mjc5ODkzZGNhOGRmNWJmNjVlZjQ0MjMxNDM4MDIyO Tg1MmRjMGJlNGNkYTN kYTQ4MzMxZjRlN2FiNjI3MjE1NGl3MTA3ZDQ4NWRmYzMyOTUzZ jZIMjg3NjQ1OGQ4MTA3N TU2N2ZkN2ZjYzUzYzYyMjFiODAwN2VkM2U5MTZiNDY2MmM2ZTV lYjQ2YzI0OGQ2ODUxNW YkMjI2MWVlNDAYOGlOMThkMTdhNTY1YzIxMDgyOGZIM2IwZWZj MDgwM2U4MzNlNDg1OD UxZTc4ODc1MTY2M2I0NjU5ZjBhZjVhNjk0OTIhNGExOThmYWVl NmFIZjlyNmMwZDA3M0M0 NjJkZDhhMml4ZmRhYjc3NmZlNDFlODkyYjBhYjY3MDQ1OGVIMj dhYmUwZTlyNGlxYmQyZDIz ZjliM2E5ZGQ5NmNhZDQxOTM4NTI0Mjc3MzBlOWEwZWE1Njk3Yj gxY2VlNTQ1OWULnoiVd ULljTEuDJqneOGMRHesi8PTnZjO2yqbmKbFKlPjwMhe7FUhFAOw74S+H+P okOREo5XhdP+ y9 /</p> <p>GuI3juYTvrIEOxGx2OsSfNS5kfRXH1DaTnb7Oyufe9r6mMIQ6 e6E0SRUIdu6YVupz0hhgd4Dof</p> <p>SBbFR3OvgOS+pUxDYgmEOrr/RA+fYi47tuHQMh+dynZqQspNdmRUmkjEpFqF03sPHS/10injqo e1Gsf8+xn52XE2qW/dnU+4XjWn/iSVNAjv2nsL+s2TG1IHbgocmpQoxy0B0SXPcRv/+2JekV37 k1XyONZkSYH+DV3aWYPXt+ym+wG0XNTqPHIU1JWAZqI2NK/cSXt9DMtCtcb8czRj6G9IXvJ9 Eny7iO6xPd9BGio9M+3QuUkZHEmJiAvgvB6R/X/3whBqk6zMHQLfo+VJcX9umW5mRtgCjzS PW6lzzFCGt/B4SK4PxT52ZC0B2kWD8VMYnffrlsTG4XUesgx47Nd5xML8w5pj/fZwKNK+EfkIP ==Z1ow29A9N3uLIXBX52LhOyji1iqfJ2FNR7AIONSEjwKoggVmkxDiuGaQi+TurpxBgat1g</p> <p>### End ASRAR El Mojahedeen v2.0 Encrypted Message ###</p> |  | <p>لوحة التحكم</p> <p>الإعدادات واخيرات</p> <p>تفعيل التوقيع</p> <p>النموذج: 1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64/65/66/67/68/69/70/71/72/73/74/75/76/77/78/79/80/81/82/83/84/85/86/87/88/89/90/91/92/93/94/95/96/97/98/99/100/101/102/103/104/105/106/107/108/109/110/111/112/113/114/115/116/117/118/119/120/121/122/123/124/125/126/127/128/129/130/131/132/133/134/135/136/137/138/139/140/141/142/143/144/145/146/147/148/149/150/151/152/153/154/155/156/157/158/159/160/161/162/163/164/165/166/167/168/169/170/171/172/173/174/175/176/177/178/179/180/181/182/183/184/185/186/187/188/189/190/191/192/193/194/195/196/197/198/199/200/201/202/203/204/205/206/207/208/209/210/211/212/213/214/215/216/217/218/219/220/221/222/223/224/225/226/227/228/229/230/231/232/233/234/235/236/237/238/239/240/241/242/243/244/245/246/247/248/249/250/251/252/253/254/255/256/257/258/259/260/261/262/263/264/265/266/267/268/269/270/271/272/273/274/275/276/277/278/279/280/281/282/283/284/285/286/287/288/289/290/291/292/293/294/295/296/297/298/299/300/301/302/303/304/305/306/307/308/309/310/311/312/313/314/315/316/317/318/319/320/321/322/323/324/325/326/327/328/329/330/331/332/333/334/335/336/337/338/339/340/341/342/343/344/345/346/347/348/349/350/351/352/353/354/355/356/357/358/359/360/361/362/363/364/365/366/367/368/369/370/371/372/373/374/375/376/377/378/379/380/381/382/383/384/385/386/387/388/389/390/391/392/393/394/395/396/397/398/399/400/401/402/403/404/405/406/407/408/409/410/411/412/413/414/415/416/417/418/419/420/421/422/423/424/425/426/427/428/429/430/431/432/433/434/435/436/437/438/439/440/441/442/443/444/445/446/447/448/449/450/451/452/453/454/455/456/457/458/459/460/461/462/463/464/465/466/467/468/469/470/471/472/473/474/475/476/477/478/479/480/481/482/483/484/485/486/487/488/489/490/491/492/493/494/495/496/497/498/499/500/501/502/503/504/505/506/507/508/509/510/511/512/513/514/515/516/517/518/519/520/521/522/523/524/525/526/527/528/529/530/531/532/533/534/535/536/537/538/539/540/541/542/543/544/545/546/547/548/549/550/551/552/553/554/555/556/557/558/559/560/561/562/563/564/565/566/567/568/569/570/571/572/573/574/575/576/577/578/579/580/581/582/583/584/585/586/587/588/589/590/591/592/593/594/595/596/597/598/599/600/601/602/603/604/605/606/607/608/609/610/611/612/613/614/615/616/617/618/619/620/621/622/623/624/625/626/627/628/629/630/631/632/633/634/635/636/637/638/639/640/641/642/643/644/645/646/647/648/649/650/651/652/653/654/655/656/657/658/659/660/661/662/663/664/665/666/667/668/669/670/671/672/673/674/675/676/677/678/679/680/681/682/683/684/685/686/687/688/689/690/691/692/693/694/695/696/697/698/699/700/701/702/703/704/705/706/707/708/709/710/711/712/713/714/715/716/717/718/719/720/721/722/723/724/725/726/727/728/729/730/731/732/733/734/735/736/737/738/739/740/741/742/743/744/745/74</p> |
|--|--|---|

# Boolean Equations

- Basic fingerprints can also use Boolean equations:

```
appid('voip/sip/IMS', 6.0, wireshark='sip') =  
(('via: sip' or 'v: sip') and 'cseq:' and (  
  'p-access-network-info:' or  
  'p-called-party-id:' or  
  'p-charging-vector:' or  
  'p-charging-vector-addresses:' or  
  'p-media-authorization:' or  
  'security-verify:' or  
  'proxy-authorization:' and 'scscf' or  
  'path:' and 'pcscf' or  
  'path:' and 'scscf'  
));
```



# Regular Expressions

- And Regular Expressions

**fingerprint('encryption/mojahedeen2')=**

**/(?:Begin|End).ASRAR.El.Mojahedeen.v2\..\{0,5}Encrypted.Message/ or**

**/Mojahedeen.v2\..\{0,5}Encrypted.Message/ or**

**/(?:Begin|End).Al-Ekhlaas.Network.ASRAR.El.Moujahedeen.V2/ or**

\* Regular expressions must include a fixed "anchor" meeting the minimum keyword length.

Bad: **/[A-Z]{3}-[0-9]{3,5}/**

OK: **/ABC-[0-9]{3,5}/**

# Binary Patterns

- And Binary Patterns

`fingerprint('botnet/IO/XXPW0023') =`

`$http and`

`'\x53\x53\x48\x00\x00\x00\x00\x00'c and`

`'\x00\x00\x00\x00\x03\x00\x53\x4D\x52'c;`

`fingerprint('botnet/IO/XXPW0023') =`

`$http and`

`hex('5353480000000000') and`

`hex('00000000300534D52');`

# Positional Logic

**fingerprint('botnet/IO/XXPW0023') =**

**pos('\x53\x53\x48\x00\x00\x00\x00\x00') == 4**

**and pos('\x00\x00\x00\x00\x03\x00\x53\x4D\x52') == 24;**

**fingerprint('botnet/IO/XXPW0023') =**

**\$http and**

**(pos('\x53\x53\x48\x00\x00\x00\x00\x00') >= 144 and  
pos('\x53\x53\x48\x00\x00\x00\x00\x00') <= 184) and**

**(pos('\x00\x00\x00\x00\x03\x00\x53\x4D\x52') >= 164 and  
pos('\x00\x00\x00\x00\x03\x00\x53\x4D\x52') <= 204);**



# When that's not enough...

- For example, take the first scenario:  
“I want to look for documents from Iran that mention a banned item”
- Just using keywords with Boolean equations, how could we restrict the term to only a document body and only coming from Iran?

# Context Sensitive Scanning

- X-KEYSCORE's context sensitive scanning engine allows you to explicitly say where you want a term to hit.
- As an early example, the Tech Strings in Documents capability allowed analysts to restrict terms to only Email, Chat or Documents Bodies
- The full XKS Context Sensitive Scanning engine allows for over 70 unique contexts to be used as part of an fingerprint

# Context Sensitive Scanning

- For example, take the first scenario:  
“I want to look for documents from Iran that mention a banned item”
- Using the XKS context for Country Code (based on NKB information) and the XKS context for Document Bodies, this easily becomes:

```
fingerprint('demo/scenario1') =  
    cc('ir') and doc_body('banned item')
```



# Context Sensitive Scanning

- As another example, let's say we want to tag all Iphone usage
- Using the XKS context for User Agent this easily becomes

```
fingerprint('demo/scenario2') =  
    user_agent('iphone');
```

# USSID18/HRA Considerations

- XKS Fingerprints may not be USSID18 or HRA compliant if they are queried on by themselves
- For example, we may want to fingerprint the use of mobile web devices like the iPhone, so that attribute could be used as part of a more complex query.
- But querying for the iPhone fingerprint itself would be a USSID18 and HRA violation.

# USSID18/HRA Considerations

- But if you want to look for an iPhone user from an Iranian Proxy accessing his Mail.ru account:

IP Address:

78. [REDACTED]

Either



AppID  
(+Fingerprints) [[fulltext](#)]:

## Field Builder

### AppID (+Fingerprints)

browser/cellphone/iphone



Add to Field

Close

## Field Builder

### AppID (+Fingerprints)

mail/webmail/mailru



mail/webmail/mailru

mail/webmail/mailru/attachment

mail/webmail/mailru/post



# Context Sensitive Scanning

What contexts are available for use in XKS Fingerprints?

# HTTP Activity Contexts (1 of 2)

html_title(expr)	The normalized extracted text web page titles <a href="#">html_title('how to' and 'bomb')</a>
http_host(expr)	The "Host:" name given in the http header. <a href="#">http_host('yahoo.com')</a>
http_url(expr)	Every URL from HTTP GET and POST commands. <a href="#">http_url('/mail/inbox?action=delete')</a>
http_url_args(expr)	All arguments given as part of a URL (ie. all text following the '?' in a URL string) <a href="#">http_url('action=delete')</a>
http_referer(expr)	The "Referer:" URL given in the HTTP header <a href="#">http_referer('http://badwebsite/cp?action=show')</a>
http_language(expr)	The normalized two letter iso-6393 language code as inferred from any http and or html header info <a href="#">http_language('fa' or 'de')</a>

# HTTP Activity Contexts (2 of 2)

http_cookie(expr)	The “Cookie:” field given in the http header. <a href="#">http_cookie(/PREF=\d\d[a-z]/)</a>
http_server(expr)	The “Server:” type name in the http header. <a href="#">http_server('GWS/2.1' or 'Apache')</a>
http_user_agent(expr)	The “User-Agent:” field given in the http header. <a href="#">http_user_agent(/Mozilla\[/45]/ or 'Chrome')</a>
web_search(expr)	The normalized extracted text from web searches <a href="#">web_search('ricin' or 'plague')</a>
x_forwarded_for(expr)	The X-Forwarded For IP address from the HTTP Header <a href="#">x_forwarded_for('1.2.3.4')</a>



# Protocol Contexts 1 of 2

ip(expr)	The source or destination IP address of the session <code>ip('127.0.0.1')</code>
from_ip(expr)	The source IP address of the session <code>from_ip('127.0.0.1')</code>
to_ip(expr)	Every URL from HTTP GET and POST commands. <code>to_ip('127.0.0.1')</code>
ip_subnet(expr)	IP subnet in CIDR notation. <code>ip_subnet('7.211.143.148/24')</code>
port(expr)	The source or destination TCP or UDP port number. <code>port('22')</code>
from_port(expr)	The source TCP or UDP port number. <code>from_port('22')</code>
to_port(expr)	The destination TCP or UDP port number. <code>to_port('22')</code>

# Protocol Contexts 1 of 2

cc(expr)	The country (either to OR from) based on IP address cc('ir' or 'pk')
from_cc(expr)	The source country based on IP address from_cc('ir' or 'pk')
to_cc(expr)	The destination country based on IP address to_cc('ir' or 'pk')
protocol(expr)	The textual form of the IP next protocol. protocol('TCP')
next_protocol(expr)	The textual form of the IP next protocol. ip_next_protocol('17')
mac_address(expr)	The MAC address of the target network device. mac_address('00:16:3E:3F:BD:EF')



# Communication Based Contexts

email_body(expr)	The UTF-8 normalized text of all email bodies. <code>email_body('how to' and 'build' and ('bomb' or 'weapon'))</code>
chat_body(expr)	The UTF-8 normalized text of all chat bodies. <code>chat_body('how to' and 'build' and ('bomb' or 'weapon'))</code>
document_body(expr)	The UTF-8 normalized text of the Office document. – Office documents include (but are not limited to) Microsoft Office, Open Office, Google Docs and Spreadsheets. <code>document_body('how to' and 'build' and ('bomb' or 'weapon'))</code>
calendar_body(expr)	The UTF-8 normalized text of all calendars. An example is Google Calendar. <code>calendar_body('wedding')</code>
archive_files(expr)	Matches a list of files from within an archive. For example is a ZIP file is transmitted, all names of files within are passed to this context. <code>archive_files('bad.dll' or 'virus.doc')</code>
http_post_body(expr)	The UTF-8 normalized text HTTP url-encoded POSTs. <code>http_post_body('action=send' and 'badguy@yahoo')</code>



# Communication Based Contexts

## Aliases

doc_email_body(expr)	This covers the email_body and document_body contexts doc_email_body('how to' and 'build' and ('bomb' or 'weapon'))
communication_body(expr)	This covers the email_body, document_body and chat_body contexts chat_body('how to' and 'build' and ('bomb' or 'weapon'))

# Context sensitivity

Why use context-sensitive scanning?

- More intuitive - you can say what you mean
- More accurate - if 'maps.google.com' is mentioned in a blog post, you don't want to try processing it as a Google Maps session
- Better performance for XKEYSCORE

# Examples

- “I want to look for people doing web searches on Jihad from Kabul”
- Using the `from_city()` and `web_search()` context this becomes

```
fingerprint('demo/scenario3') =  
    from_city('kabul') and web_search('jihad');
```



# Examples

- “I want to look for people using Mojahdeen Secrets encryption from an iPhone”
- You can even use existing fingerprints in a fingerprint definition! So this becomes:

```
fingerprint('demo/scenario4') =  
    fingerprint('encryption/mojahdeen2' and  
    fingerprint('browser/cellphone/iphone')
```

# Examples

- “I want to look for documents containing this regular expression”
- Using doc\_body this becomes:

```
fingerprint('demo/scenario5') =  
    doc_body(/blah[a-z]{3-5}something/)
```

## Example 4

- “I want to look for E-mails that mention words from various categories of interest to CP”
- You can use multiple variables in an equation like this:

```
topic('wmd/acw/govtorgs') =  
    email_body($acwitems and $acwpositions and  
    ($acwcountries or $acwbrowsers or $acwports));
```



# Example 4

- **\$acwitems** = 'machine gun' or 'grenade' or 'AK 47'
- **\$acwpositions** = 'minister of defence' or 'defense minister'
- **\$acwcountries** = 'somalia' or 'liberia' or 'sudan'
- **\$acwbrowsers** = 'south africa' or 'serbia' or 'bulgaria'
- **\$acwports** = 'rangoon' or 'albasra' or 'dar es salam'

```
topic('wmd/acw/govtorgs') =  
    email_body($acwitems and $acwpositions and  
    ($acwcountries or $acwbrowsers or $acwports));
```

# Advanced Code-Based Fingerprints

- What happens when there are no keywords or regular expressions that will help identify the traffic of interest to you?
- As enough example, many of the CT Targets are now smart enough to not leave the Mojahedeen Secrets header in the E-mails they send. How can we detect that the E-mail (which looks like junk) is in fact Mojahedeen Secrets encrypted text
- A C++ code fingerprint can help evaluate that data



# Code Based Fingerprint

The screenshot shows a web browser window with a message interface. The main content area displays a large block of Base64-encoded text, which is a code-based fingerprint. The text is as follows:

```

r/RgTzT/ATRhn2E1Zjg1OWQyNWRjMmE2ZTdlNzZmZDh1ODUxZWZhMDQ1MjYwMjYzGU0
ZGYwMjckMmJmNTA4ZDY2Yjk0MGU2NGNiYjg6MzNjZTc6MThjY2Y1ZmY6MTgzZDlkYjhjMTE
xOGYzYjclZDdlMDAxNTQzZmVINDVIY2YyMGUjYjU2ODkyYjdmYjFjYjAzMWM5ZDQ2OWFIMzg
4NThhM2I1Mjc5ODkzZGNhOGRmNWJmNjVIZjQ0MjMxNDI4MDIyO Tg1MmRjMGJiNGNkYTN
kYTQ4MzMxZjRlN2FmNjI3MjE1NGI3MTA3ZDQ4NWRmYzMyOTUzZjZIMjg3NjQ1OGQ4MTA3N
TU2N2ZkN2ZjYzUzYzYyMjFjODAwN2YkM2U5MTZiNDY2MmM2ZTViYjQ2YzI0OGQ2ODUxNW
YkMjI2MWMVINDAyOGI0MThkMTdhNTY1YzIxMDgyOGZIM2IwZWZjMDgwM2U4MzNINDg1OD
UxZTc4ODc1MTY2M2I0NjU5ZjBhZjVhNjk0OTIhNGExOThmYWVlNmFIZjlyNmMwZDA3MDM0
NjJkZDhhMmI4ZmRhYjc3NmZINDFkODkyYjBhYjY3MDQ1OGVIMjdhYmUwZTlyNGIxYmQyZDIz
ZjliM2E5ZGQ5NmNhZDQxOTM4NTI0Mjc3MzBIOWEwZWE1Njk3YjgxY2VINTQ1OWULnoiVD
ULijTEuDJqneOGMRHesi8PTnZjO2yqbmKbFkIPjwMhe7FUhFAOw74S++P0kOREo5XhdP+ y9
/
Gu3juYTvriEOxGx20sSfNS5kfrXXH1DaTnb7Oyufe9r6mMIQ6
e6E0SRUldU6YYupz0hhgd4Dof
SBbFR3OvgOS+pUxDYgmEOrrA+fYi47tuHQMh+dynZqQspNdmRUmkjEpFqF03sPHS/10injoo
e1Gsf8+xn52XE2q/WdnU+4XjWnl/isVNAjv2nsL+s2TG1IHbgocmpQoxy0B0SXPcRv/+2JekV37
k1XyONZk9YH+DV3aWYPXt+ym+wG0XNTqPHIU1JWAZql2NK/cSXt9DMtCtcb8czRj6G9IXvJ9
Eny7t06xFd9BGio9M+3QuUkZHEmJiAvgvB6R/X3whBqk6zMHQLfo+VJcX9umW5mRtgCjzS
PW6lzzFCGvB4SK4PxT52ZC0B2kWD8VMyNffrlsTG4XUesgx47Nd5xML8w5pjY7ZwKNK+EflKIP
==Z1ow29A9N3uLIXBX52LhOyj1iqfJ2FNR7AIONSEjwKoggVmKxDiuGaQi+TurpxBgat1g

```

The interface includes a header with the date "Mar 2007" and a sidebar with various navigation links. The bottom status bar indicates "Displaying 1 items" and "Hidden fields".



# Advanced Code-Based Fingerprints

```

fingerprint('encryption/mojaheden2/hidden44') =
$moj_cipher_first_test
: C++ Extractors : {{
msg1 = /([a-zA-Z0-9\+\-\/]{6}[DHLPTX3bfjnrvs37\+\/]{4-9\+\-\/})AT([ABEFIJMNQFUVVYZedghkl][hijklmxyz012345][MNOYZ][DGTWjwz2]{12})/c;
msg2 = /([a-zA-Z0-9\+\-\/]{6}[DHLPTX3bfjnrvs37\+\/]{4-9\+\-\/})AK([ADEFIJMNQFUVVYZedghkl][hijklmxyz012345][MNOYZ][DGTWjwz2]{12})/c;
}}
main : {{
std::string msg;
if (msg1)
    msg = msg1[0];
else if (msg2)
    msg = msg2[0];
else
    return false;

char buf[16];
char chunk1[16];
char chunk2[16];
char chunk3[16];
if(true){
    snprintf(chunk1, 16, "%02x%02x%02x%02x",
        msg[10] & 0xff,
        msg[11] & 0xff,
        msg[12] & 0xff,
        msg[13] & 0xff);
    snprintf(chunk2, 16, "%02x%02x%02x%02x",
        msg[14] & 0xff,
        msg[15] & 0xff,
        msg[16] & 0xff,
        msg[17] & 0xff);
    snprintf(chunk3, 16, "%02x%02x%02x%02x",
        msg[18] & 0xff,
        msg[19] & 0xff,
        msg[20] & 0xff,
        msg[21] & 0xff);
    if(!((strcmp(chunk1, chunk2) == 0) ||
        (strcmp(chunk2, chunk3) == 0) ||
        (strcmp(chunk1, chunk3) == 0))){
        std::string msg_decoded = xks::base64dec(msg);
        for(size_t i = 7; i < msg_decoded.size(); i++){
            if(msg_decoded[i] < '0' || (msg_decoded[i] > '9' && msg_decoded[i] < 'a') || msg_decoded[i] > 'f'){
                return false;
            }
        }
    }
    snprintf(buf, 16, "%02x%02x%02x%02x",
        msg_decoded[1] & 0xff,
        msg_decoded[2] & 0xff,
        msg_decoded[3] & 0xff,
        msg_decoded[4] & 0xff);
    std::string keyid_hex = buf;
}

```

## Field Builder

### AppID (+Fingerprints)

# Advanced Code-Based Fingerprints

- As another example, some of the activity from the Conficker botnet simply can't be detected with keywords or regular expressions
- In cases like this, C++ code can be used inside a fingerprint to test the data further

# Advanced Code-Based Fingerprints

```

fingerprint('botnet/conficker_p2p_udp_data', 7.0) =
$udp and not 'to ap 00' and not 'fm ap 00'
: c++()
// Classification: CONFIDENTIAL//REL TO USA, FVEY
// NOT releasable to third-parties
uint8_t key8;
uint8_t key9;
uint8_t pkt_type;
uint0_t decrypted_bytes[4];
uint32_t running_hash = 0;
uint32_t K_high;
uint32_t K_low;
uint32_t stored_hashes[4] = {0,0,0,0};
uint32_t min_pkt_len;
uint32_t max_pkt_len;
uint32_t t;
packet_t pkt;

while(pkt = get_packet())
{
    if (pkt.size < 10)
        return false;

    key8 = (uint8_t)(pkt.data[7]<<1 | (pkt.data[7]>>7)&1);
    key9 = (uint8_t)(key8<<2 | ((pkt.data[7]>>5)&1));

    if (((key9 ^ pkt.data[9]) & 0x80) != 0x80)
        return false; // Not Conficker, so abort
    if ((key9 ^ pkt.data[9]) & 0x02)
        return false; // bit not set for UDP packets

    if (pkt.size < 23)
        continue;

    if ((key8 ^ pkt.data[8]) != 0x80)
        continue;

    pkt_type = (key8 ^ pkt.data[8]) >> 3;

    if (pkt_type & 0x10) // summary
        continue;

    if (!(pkt_type & 0x08)) // not a data packet
        continue;

    min_pkt_len = 23;
    max_pkt_len = (uint32_t)pkt.size;

    K_high = uint32_t(pkt.data[7])<<24u | uint32_t(pkt.data[6])<<16u | uint32_t(pkt.data[5])<<8u | uint32_t(pkt.data[4]);
    K_low = uint32_t(pkt.data[3])<<24u | uint32_t(pkt.data[2])<<16u | uint32_t(pkt.data[1])<<8u | uint32_t(pkt.data[0]);

    running_hash = 0;

    for(t=1; t<max_pkt_len; t++)
    {
        if(t>=8) // decrypt data
        {

```

## Field Builder

### AppID (+Fingerprints)

botnet/conficker\_p2p\_udp\_data

Add to Field

Close



# Meta-data Extracting Fingerprints

- What happens when you find data and want some pieces of meta-data extracted?
- XKS Fingerprints can be used to extract meta-data to select XKS database tables.
- Or if no existing database is applicable, you can define your own database schema for the meta-data

# Free File Upload Sites

- As a real life example, think of all of various Free File Upload (FFU) sites of interest
- When a user uploads a document they get a response page that looks like this:

# Free File Upload Sites

## Welcome to <sup>z</sup>SHARE

With <sup>z</sup>SHARE you can upload files, images, videos, audio and flash for free. Simply use the upload form below and start sharing! You can also use <sup>z</sup>SHARE as your personal file storage: backup your data and protect your files. First Time? Read our [FAQ](#)!

- [Upload now](#)
- [Login](#)
- [Create Free Account](#)
- [Premium](#)
- [FAQ](#)

## File Uploaded

The file **khi pics.zip** was successfully uploaded! (4.04MB). You're now ready to share it with unlimited people or keep it as a backup.

Download Link

<http://www.zshare.net/download/637199570b174c9f/>

Link for forums:

[URL=<http://www.zshare.net/download/637199570b174c9f/>]

Direct Link:

<http://www.zshare.net/download/637199570b174c9f/>

Delete Link:

<http://www.zshare.net/delete.html?63719957-7c8893b1k>

[E-mail Me This Info](#)

To receive all the info on the file you uploaded, such as **removal instructions** and **download link**, enter your e-mail address on the field below:

Your e-mail:



# Free File Upload Sites

- Look at all the great information on that page:

## File Uploaded

The file **khi pics.zip** was successfully uploaded! (4.04MB). You're now ready to share it with unlimited people or keep it as a backup.

Download Link

<http://www.zshare.net/download/637199570b174c9f/>

Link for forums:

[URL=<http://www.zshare.net/download/637199570b174c9f/>]

Direct Link:

<http://www.zshare.net/download/637199570b174c9f/>

Delete Link:

<http://www.zshare.net/delete.html?63719957-7c8893b1b>

# Free File Upload Sites

- How can we quickly get that information extracted as Meta-data and be agile enough to respond to each FFU site which may have its own format
- XKS “v4” Fingerprints allow you to use the XKS Fingerprint Language to extract meta-data into the XKS database
- Fingerprints are deployed within an hour of being accepted meaning you no longer need to wait for all 130+ XKS sites to be upgraded to have the latest and greatest capabilities.

# Free File Upload Sites

```

appid('filetransfer/web/zshare_net/upload/response', 5.0) =
    http_title('zSHARE') and 'zshare.net/delete.html'
    : c++
extractors : {{
    wft_file_name = /The\sfile\s<strong><font\scolor=\ "#333333\s">{[^<]{1,300}}\s</;
    wft_delete_url = /zshare.net\/delete.html\?{[0-9]+}-{[0-9a-zA-Z]{32}}\s"/;
    wft_upload_id = /<font color=\ "#666666\s"><a href=\ "http:\\\/\\\/www\.zshare\.net\/{[^\\\/]+\\\/{[0-9]+}[0-9a-f]{8}\\\/;
    wft_url = /<font color=\ "#666666\s"><a href=\ "(http:\\\/\\\/www\.zshare\.net\/{[^\\\/]+\\\/{[^\\\/]+}\\\/;
    wft_uploader_username = /<small>Logged in as: {[^<]+}<\/small>/;
}}
main = {{
    if (wft_delete_url) {
        DB["web_file_transfer"]["wft_upload_id"] = wft_upload_id[0];
        DB["web_file_transfer"]["wft_delete"] = wft_delete_url[0]+"-"+wft_delete_url[1];

        DB["web_file_transfer"]["wft_site_name"] = "zshare.net";
        DB["web_file_transfer"]["transfer_type"] = "upload";

        if (wft_file_name) {
            DB["web_file_transfer"]["wft_filename"] = wft_file_name[0];
        }

        if (wft_url) {
            DB["web_file_transfer"]["wft_url"] = wft_url[0];
        }
        if (wft_uploader_username) {
            DB["web_file_transfer"]["uploader_username"] = wft_uploader_username[0];
        }
        DB.apply();
    } else {
        logger.debug("filetransfer/web/zshare.net/upload/response: Host regexs didn't match");
    }
    return true;
}};

```



# Meta-data Extracting Fingerprints

- All you do is tell XKS when to start extracting meta-data

```
appid('filetransfer/web/zshare_net/upload/response', 5.0) =  
    http_title('zSHARE') and 'zshare.net/delete.html'  
: c++
```

- ```
extractors : {  
  wft_file_name = /The\sfiler\sfiler<strong><font\scolor=\"#333333\">{[^<]{1,300}}\sfiler</>;  
  wft_delete_url = /zshare.net\delete.html\?{[0-9]+}-{[0-9a-zA-Z]{32}}\sfiler/;  
  wft_upload_id = /<font color=\"#666666\"><a href=\"http:\\\\www\zshare.net\/[^\\/]+\sfiler{[0-9]+}[0-9a-f]{8}/>;  
  wft_url = /<font color=\"#666666\"><a href=\"(http:\\\\www\zshare.net\/[^\\/]+\sfiler[^\\/]+\sfiler)/>;  
  wft_uploader_username = /<small>Logged in as: {[^<]+}<\sfiler/;  
}
```

# Meta-data Extracting Fingerprints

- Finally tell it which database tables you want to store the information:

```
main = {{
  if (wft_delete_url ) {
    DB["web_file_transfer"]["wft_upload_id"] = wft_upload_id[0];
    DB["web_file_transfer"]["wft_delete"] = wft_delete_url[0]+"-"+wft_delete_url[1];

    DB["web_file_transfer"]["wft_site_name"] = "zshare.net";
    DB["web_file_transfer"]["transfer_type"] = "upload";
```

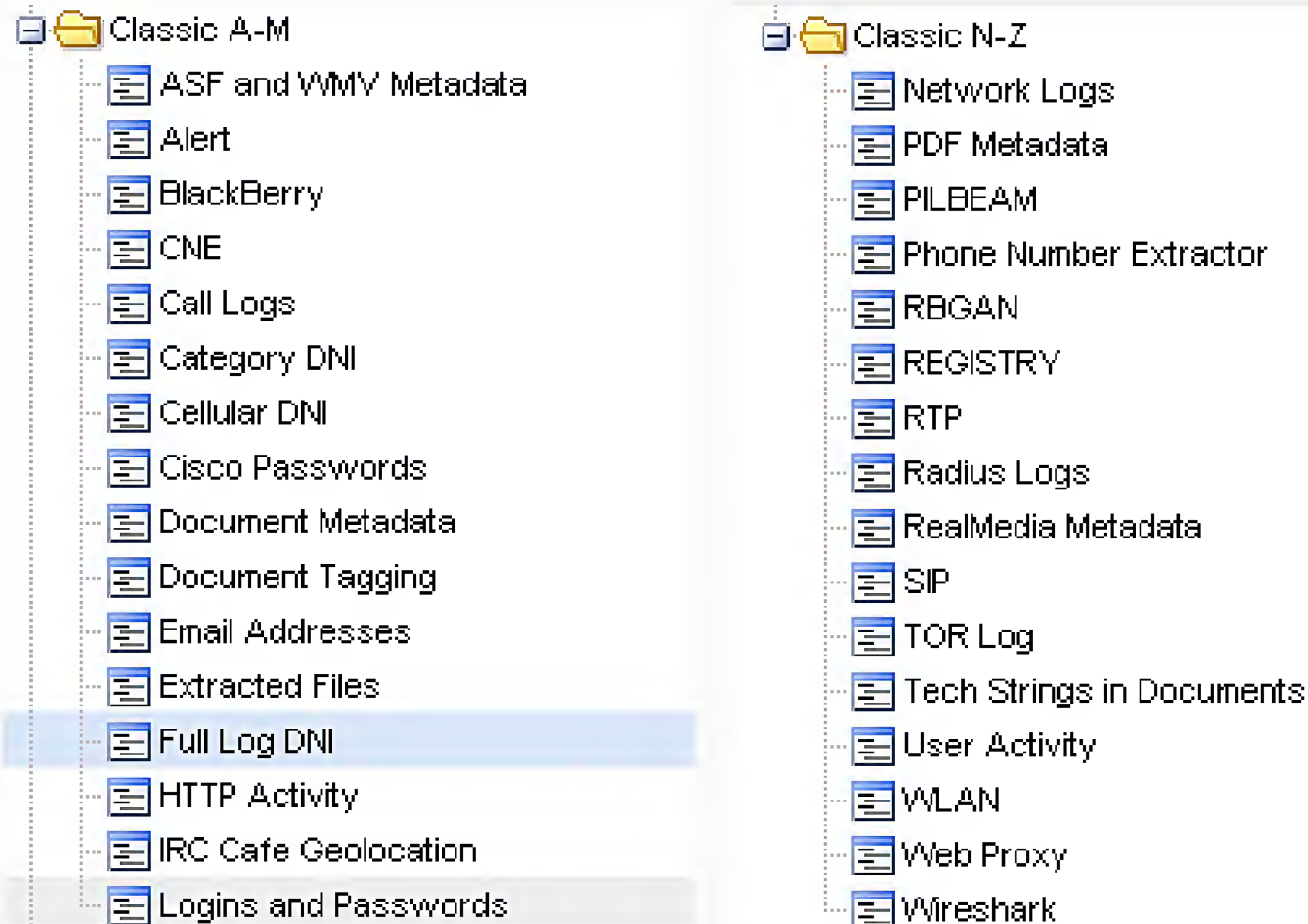
| File URL                                                                                                      | Filename     |
|---------------------------------------------------------------------------------------------------------------|--------------|
| <a href="http://www.zshare.net/download/637199570b174c9f">http://www.zshare.net/download/637199570b174c9f</a> | khi pics.zip |

| Transfer Type | Upload ID | Delete ID                        | Site Name  |
|---------------|-----------|----------------------------------|------------|
| upload        | 63719957  | 7c8893b1bf04170771dca3e7f0756a26 | zshare.net |



# Meta-data Extracting Fingerprints

- What if the meta-data you want to extract doesn't fit nicely into any of the existing XKS meta-data tables?



# Meta-data Extracting Fingerprints

- Define your own with the “Microplugin” query forms:



# Meta-data Extracting Fingerprints

- Example MS2 KeyIDs

Search: Ms2 Extract Keyids

Query Name:

Justification:

[Recent Justifications](#)

Additional Justification:

Miranda Number:

Datetime:

Start:

2010-05-03



00:00



Stop:

2010-05-04



23:59



ejKeyID:

Username<realm>:

IP Address:

From



[\[IP Address Field Builder\]](#)

IP Address:

To



[\[IP Address Field Builder\]](#)

Port:

From



Port:

To





# Meta-data Extracting Fingerprints

Search: Ccne Byzantine Raptor Trojan3

Query Name:

Justification:  [Recent Justifications](#)

Additional Justification:

Miranda Number:

Datetime:  Start:   Stop:

brt\_decrypt:

brt\_hostname:

brt\_ipaddress:

brt\_length:

brt\_osversion:

brt\_packet\_type:

brt\_sequence\_num:

brt\_username:

Username <realm>:

IP Address:  From  [\[IP Address Field Builder\]](#)

IP Address:  To  [\[IP Address Field Builder\]](#)

# New Fingerprint GUI

- New XKS Fingerprint GUI allows analysts to directly test, submit and manage fingerprints through the web

The screenshot displays the XKS Fingerprint GUI. On the left is a 'Navigation Menu' with a 'Fingerprints' folder icon and four sub-items: 'Validate / Submit', 'Approved', 'Pending', and 'My Signatures'. The main content area is titled 'Fingerprint Validation / Submittal' and features three steps: 'Step #1' with a 'Compile' button (gear icon), 'Step #2' with a 'Test Against Session Data' button (key icon), and 'Step #3' with a 'Save' button (floppy disk icon). A 'Help' button is in the top right. Below the steps is a 'Global Variable Declarations' section with a text area containing the instruction 'Type or paste any global VARIABLE DECLARATIONS here.'. This is followed by a 'Signature' section with a text area containing the instruction 'Type or paste a FINGERPRINT definition here.'. A footer bar at the bottom of the main area contains the text 'Press Compile when done editing'.

Navigation Menu

Fingerprints

- Validate / Submit
- Approved
- Pending
- My Signatures

Fingerprint Validation / Submittal

Step #1 Step #2 Step #3

Compile Test Against Session Data Save

Help

Global Variable Declarations

Type or paste any global VARIABLE DECLARATIONS here.

Signature

Type or paste a FINGERPRINT definition here.

Press Compile when done editing

# New Fingerprint GUI

- New XKS Fingerprint GUI allows analysts to directly test, submit and manage fingerprints through the web

The screenshot displays the 'Fingerprint Validation / Submittal' web interface. It features a three-step process bar at the top: Step #1 'Compile' (active, green checkmark), Step #2 'Test Against Session Data' (disabled, greyed out), and Step #3 'Save' (disabled, greyed out). Below the process bar is a 'Global Variable Declarations' section with a text area containing the code: `$test = 'bomb' or 'missile' or 'ied';`. The 'Signature' section contains a text area with the code: `fingerprint('test/test1') = email_body($test);`. A green success message 'Success!' is displayed below the signature section. The 'Results' section at the bottom shows a large green 'SUCCESS!' message, followed by the text: 'Congratulations, your fingerprint was successfully compiled!' and 'Now use the Test button to run it against the designated session data.'

Fingerprint Validation / Submittal

Step #1 Step #2 Step #3

Compile Test Against Session Data Save

Global Variable Declarations

```
$test = 'bomb' or 'missile' or 'ied';
```

Signature

```
fingerprint('test/test1') = email_body($test);
```

Success!

Results

**SUCCESS!**

Congratulations, your fingerprint was successfully compiled!

Now use the Test button to run it against the designated session data.



# Questions?

# Syntax Rules


- The definition of the fingerprint will look like this:

`fingerprint('test/blah/something', owner = [REDACTED]) =`

Note the single quotes needed for the fingerprint name and owner

# Syntax Rules

- Secondly every fingerprint definition must be completed by a semi-colon.

```
fingerprint('test/blah/something', owner = ) =  
    'badguy' ;
```



# Syntax Rules

- Variables also must be completed by a semi-colon.

```
$badguy =  
    'bomb' or 'gun' or 'weapon' ;  
fingerprint('test/blah/something', owner = '██████████') =  
    $badguy;
```

# Syntax Rules

- Definitions and Variables can span multiple lines

```
$badguy =  
    'bomb' or  
    'gun' or  
    'weapon' ;  
fingerprint('test/blah/something', owner = [REDACTED]) =  
    $badguy;
```